THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 59, No. 8, October 1980
Printed in U.S.A.

An Experimental Speech Storage and Editing Facility

By N. F. MAXEMCHUK

(Manuscript received December 7, 1979)

This paper describes an experimental system that enables text-based office services to be merged with a speech storage facility. The principal new component of this system is an editor which allows a user to modify speech messages. The discussion covers the system architecture, the facilities developed to make speech editing tractable, and the speech processing required to implement the system.

I. INTRODUCTION

The primary input device to speech storage facilities has been the standard telephone.[1-3] In an office environment, a system user may have a more sophisticated terminal capable of simultaneous data and speech. Operating a speech storage facility in an environment with data and speech makes it possible to perform operations that were previously intractable on speech messages and creates the opportunity to investigate new classes of services that merge text and speech.

This paper describes an experimental system that enables text services to be merged with a speech storage facility. The principal new component in this system is an editor that allows a user to modify speech messages. The system will be used to help determine the roles speech and text should play in an office environment. It provides researchers with the ability to develop and evaluate parallel speech and text services, to mix the two modes in order to determine what synergisms exist, and to evaluate alternative user interfaces.

The speech editor and the facilities that have been developed to make speech editing tractable are described in Section II. The system architecture permits text and speech services to be merged and allows speech services and interfaces to be developed without concern for the real-time requirements of speech sampling and reconstruction. This architecture is described in Section III. In Section IV, the capabilities

and implementation of the hardware and software developed to support this system are presented. Some speech processing functions must be performed in real time. These functions and their implementation are outlined in Section V.

II. THE SPEECH EDITOR

The objective of the speech editor is to give a user the same editing capabilities on a speech file that are available on a text file. An editor has been implemented that inserts, deletes, changes, or repositions speech segments in a speech message. The user implements these commands by marking positions in the speech message while recording or playing back a message, the beginning and end of message being marked by the system, and instructing the system to modify the message at these markers. Several techniques have been found that make speech editing more tractable. These techniques combine the text and speech capabilities of this system.

When a user establishes a marker in a speech message, he or she may associate a text descriptor with the marker. The text descriptors help the user recall the significance of the markers. For instance, if paragraphs are marked while recording a long document, the text descriptors associated with the markers may be the outline of the document. This list of text descriptors may be processed with a text editor to determine where in the speech document various topics are discussed without performing speech recognition.

To assist the user to determine the relative position of markers in a message, a time line with markers is displayed. The distance at which a marker is displayed along this line is proportional to the time at which the marker occurs in the message. After the markers are established, the user may play back the message starting at a marker, and may move the markers forward or backward in time to accurately determine the position at which an editing change will occur.

Before implementing an editing change, the user can review the change to determine what it will sound like. The review procedure plays back part of the original speech message prior to the change, the new speech segment, if one exists, and part of the original message following the change. The various speech segments during this playback operation are delineated by a prerecorded tone to show the position of the editing operation. After reviewing an editing change, the user may modify the marker positions or the new speech segment before implementing the change.

In this editor, the speech message can only be modified at silent intervals of a specified duration. By placing this constraint on the editing operations, words are not chopped when speech segments are joined, and the accuracy with which a user must position pointers is

reduced. Silent intervals occur between many words in the speech message, but not all of the words. To show the user where editing can be done and to give the user time to position a marker, a playback mode has been implemented in which all silent intervals that are long enough to perform editing are extended for a fixed duration. For instance, editing changes can be performed at pauses 1/16th of a second or longer. In this mode, all pauses greater than or equal to 1/16th of a second are played back for one second.

III. SYSTEM

The speech storage system, Fig. 1, has two principal components, a main computer system, on which all office services are implemented, and a speech-storage-system computer, which transfers speech samples between a telephone line and a mass storage medium. A user on the system has two communications paths, a control or data path to the main computer, and a speech path to the speech storage system computer. These paths are established through the switched telephone network. In the present configuration, the switch is the XDS[4] switch. The main computer can instruct this switch to connect a user's telephone to an input port of the speech-storage-system computer. This allows the speech storage system ports to be shared and enables the system to initiate the delivery of speech messages to the users.

The user interface and services in the speech storage system are implemented on the main computer. This computer translates a user's input requests into a sequence of primitive commands for the speech storage system computer. The text-based office services are also implemented on the main computer. This configuration has several advantages: text and speech services can be merged, speech services and user interfaces can be modified without regard to the real-time constraints of speech sampling and reconstruction, and the utilization of parallel speech and text services, such as a message system with both speech and text messages, can be monitored.

Certain high-level system tasks related to speech messages have been delegated to the main computer. These tasks include security, recovery after a computer fault, and archiving. This simplifies the speech-storage-system computer operating system by taking advantage of functions which must be performed by the main computer to support text services. When the speech-storage-system computer stores a speech message, it must provide the main computer with enough information to retrieve this message. The disk memory management system used by the speech-storage-system computer allows an entire speech message to be retrieved given a pointer to the beginning of the message. After a computer failure, the main computer is able to tell the speech-storage-system computer all the disk sectors

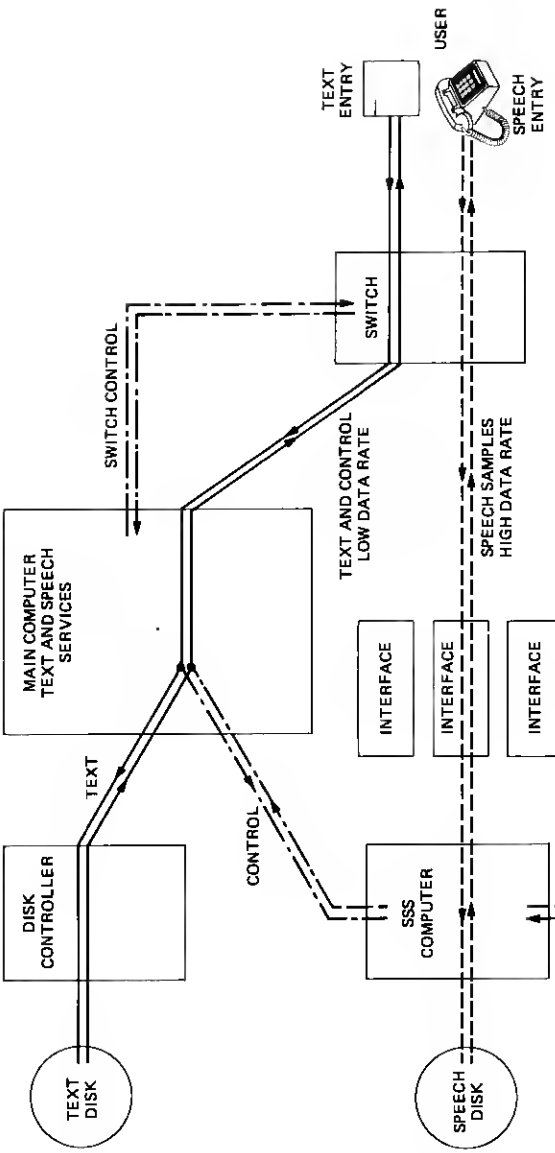

Fig. 1—Speech-storage-system architecture. The user's interface consists of a text and a speech device. The text device is switched to the main computing facility, on which services are implemented, and the speech device is switched to a speech-storage-system computer, which controls a disk used for storing speech samples. The main computer controls the speech storage computer as a peripheral and determines the operations that will be performed on the speech messages.

that contain valid speech messages by sending the speech-storage-system computer a list of the beginnings of messages. Since the main computer is responsible for tracking speech messages, it is able to restrict the users who have access to these messages in the same way it restricts the users who have access to a text file. Therefore, the speech messages are as secure as text messages. In addition, the same data base which the main computer uses to archive text messages can be used to archive speech messages.

The speech-storage-system computer is a peripheral device for the main computer. It passes speech samples between a disk storage device and a telephone line under the control of the main computer. In this configuration, the speech-storage-system computer prevents the main computer from being loaded by the input-output requirements of speech sampling since a relatively small amount of information, pertaining to the control of the speech signal, passes through the main computer. The speech-storage-system computer satisfies the real-time constraints of sampling and reconstructing the speech signal during the record and playback operations, it tracks the positions on the disk at which the speech samples in a message are stored, and it determines which disk positions are available for storing newly recorded messages. This computer processes received speech messages in real time to determine where silent intervals occur, and run length encodes these intervals. It is also capable of processing stored speech messages in nonreal time and writing the processed message to the disk. This capability has been used to evaluate speech processing algorithms, and may be used to implement speech compression algorithms.

The main computer instructs the speech storage system computer to record or playback a speech message by issuing primitive commands. These primitive commands contain parameters which specify the features of the operation to be performed. During these operations, the main computer may issue additional commands to the speech-storage-system computer to implement requests from the user. To edit a message, the main computer first verifies the validity of the user's commands, then issues a sequence of commands to the speech-storage-system computer. Verifying an editing command consists of determining that the proper number of markers and newly recorded segments have been established and that the markers are in the proper relative time sequence. For instance, the second marker in the delete command must occur after the first. The same information that is used to construct the time line is used to verify the relative position of markers.

When a command is issued to play back a message, the parameters specify the operations to be performed on the silent intervals and the playback rate. The maximum silent interval can be limited, all silent intervals between certain bounds can be eliminated, and all silent

intervals can be increased by a specified multiplier. The main computer uses these parameters to make normal playback less tedious, modify the playback rate, and facilitate editing. Additional parameters can be specified to play back the message from tbe beginning or from an intermediate point in the message, and to stop the message at a point other than the end. These options are used to review editing changes before implementing them, to determine wbere pointers occur, and to play back only part of a long speech document.

While playing back a message, the main computer can instruct the speech-storage-system computer to pause, resume, stop, get a pointer to the present position, increase or decrease the playback rate, or jump forward or backward by a specified time or to a silent interval of a specified duration. Jumping forward or backward in time allows the user to "thumb through" a long speech message until the desired area in a message is found, or to replay part of a message he has just heard.

The system stops recording a message if the recording time exceeds a maximum value, if an extended silent interval occurs, or upon command from the user. The parameters issued by the main computer with the record command specify the maximum record duration and the maximum tolerable silent interval. Another parameter from the main computer indicates whether silent intervals are run-length-encoded or whether all intervals are recorded. During normal operation, silent intervals are run-length-encoded. However, to evaluate certain speech processing algorithms, it is necessary to store silent intervals. While recording, the main computer can issue commands to pause, stop recording, resume recording, or get a pointer to the present position in the message.

The speech-storage-system computer performs two functions on a message that allows the various editing functions to be implemented. It can divide a message at a specified pointer into two messages, and it can join two messages together. To insert a speech segment at a particular point in a speech message, the main computer instructs the speech storage system computer to divide the original message at the pointer, to join the speech segment to the first part of the original message, and to join the second part of the original message to this combination. If the pointer is at the beginning or end of the original message, the main computer does not request the divide operation, and only one join operation is required. Similar sequences of divide and join operations are used to implement the other editing commands.

IV. SPEECH-STORAGE-SYSTEM COMPUTER

The objective of the speech-storage-system computer system is to simultaneously provide the services described in Section III for several users. In the present configuration, the speech signal is an 8-bit/

sample, μ-law companded signal with 8K samples/second. The speech-storage-system computer is a Digital Equipment Corporation LSI 11/02, and the storage medium is a 20-megabyte disk with a Xylogics disk controller. The system is capable of supporting three simultaneous users and storing up to 40 minutes of uncompressed speech. With silent intervals compressed, it is capable of over an hour of recording.

The only change in standard computer hardware needed to implement this system is in the interrupt structure of the direct memory access unit (DMA) between the telephone line and the speech-storage-system computer. Instead of transferring a specified amount of information, interrupting the processor and stopping, the modified DMA continuously transfers speech samples between an area in the computer memory and the phone line. When the DMA reaches the end of the specified memory area, it starts again from the beginning. It generates an interrupt each time it is half-way through and completely through the allocated memory. Using this device, the speech-storage-system computer need not reinitialize the DMA within a sample period of the speech signal. Instead, it must only process the speech samples in half the allocated memory before the DMA completely accesses the samples in the other half. Therefore, the larger the allocated memory, the greater the permissible variance in processing time for this and the other processes occupying the speech-storage-system computer. In the present system, 4096 bytes of memory are allocated for each DMA. This corresponds to four interrupts per second.

The software system in the speech-storage-system computer is an interrupt-driven, message-oriented system. Interrupts occur when the operations performed by the DMA or the disk are completed and when characters are received from the main computer or the console. Messages are generated when an end-of-line character is received from the computer or console, whenever DMA interrupts occur, and by certain software programs. Each message has a list of functions associated with it, and each function has a priority. The functions determine what effect the message will have on the system and what will be done with the data associated with a message. The priorities reflect the importance of the operations to be performed and the real time constraints of the system (in Table I, 1 = highest priority).

The system has a queue corresponding to each priority. Each hardward device has its own queue. When a message is first generated, it is placed in a queue corresponding to the priority of the first function in its list, and each time a function is completed the message is placed in the queue specified by the priority of the next function in its list. This allows high-priority tasks to pre-empt message processing at specified intervals. When the central processor (CPU) completes a function, it examines the queues, starting from the highest priority

Table I—Speech-storage-system priorities

Priority	Functions
1	Disc input/output requests (hardware)
2	Control of memory management systems
3	All other functions in list to process speech samples (real-time speech processing)
4	First function in list to process speech samples (real-time speech processing)
5	Output to speech-storage-system console (hardware)
6	Output to main computer (hardware)
7	Messages from main computer (commands), functions related to deleting messages, nonreal-time speech processing.

queue, until it finds a nonempty queue, and executes the next function on this message's list. The queues corresponding to hardware devices are skipped when the device is busy. This allows the hardware devices and the CPU to operate simultaneously. For instance, if the disk is presently transferring data between the disk and computer memory, the CPU will not examine the queue of messages which are to access the disk. Instead, it will determine if any lower priority functions are to be performed. If all the queues are empty, the CPU continues examining the queues until a message appears due to one of the interrupt mechanisms.

Each recorded message has its own disk memory management system. The memory management system contains control sectors and storage sectors. The storage sectors are disk sectors in which the speech samples are stored, and the control sectors are disk sectors with lists of storage sectors. The control sectors in a message are linked to the previous and next control sectors in the message. This arrangement allows a speech message to have any length up to the total disk capacity. The linkage in both directions allows the system to move forward or backward in time during playback. When the system is reinitialized after a computer failure, all the disk sectors in a recorded message can be determined from a single control sector in the message.

The control sector contains a parameter associated with each storage sector. This parameter is used in the rate conversion techniques to allow a decision to play back a sector to be made without reading the sector from the disk. Silent intervals are stored in the control sector by placing a unique word in the sector position and the number of silent sectors in the parameter position. This allows silent sector lengths to be modified easily.

Markers in the speech message consist of a control sector number and a displacement which points to a storage sector. The editing operations are primarily operations on the control sectors of the message. Two messages are joined by linking the last control sector of one message to the first control sector of another. A message is divided at a marker by creating a new control sector which is linked to the

control sector following that at which the division is to occur, and by putting the storage sectors following the pointer displacement in this control sector. The original control sector is then disengaged from the control sector it was linked to, and all storage sectors following the pointer displacement are removed from the original control sector.

V. SPEECH PROCESSING

In real time in the speech-storage-system microcomputer, silent intervals are detected, the playback rate is modified, and silent intervals are restored for playback. The real-time capability was the primary constraint on the class of algorithms considered to perform these functions. Difficulty in describing an adequate criterion function has curtailed a formal optimization of these algorithms. However, the nonreal-time processing capabilities of the speech storage system has been used to implement and compare a large number of algorithms. The most promising of these are described in this section.

The silent interval detector is central to many real-time processing functions in the speech storage system. It is used to reduce disk storage, modify the playback rate, and determine permissible editing positions. However, even a simple energy calculation on the received samples is beyond the real-time capabilities of the speech storage system microcomputer. The basic unit of duration for silent intervals in this system is 512 samples, approximately 1/16th second. This is one storage sector on the disk. In the speech storage system, if a received block is deemed to be silent, it is added to a run length count; if not, it is stored.

It has been found that, if as few as 32 of the 512 samples in a sector are selected pseudo-randomly, the number of these samples which exceed a particular value provides an adequate indication whether or not the energy in the sector exceeds a threshold, T. In the TASI system,[5] silent interval detectors are used in which the acquisition time for an active signal is shorter than the release time. It is necessary to adopt a similar strategy here, to prevent unvoiced sounds within words from being interpreted as silent intervals, the beginnings and ends of words from being clipped, and noise within silent intervals from being interpreted as speech. Low-energy sectors within words and at the end of words are retained by requiring L successive silent sectors before transferring the signal from an active state to a silent state. By making the number of times the threshold must be exceeded to declare a sector active larger when the signal is in a silent state than when it is in an active state, noise between words is prevented from transferring the signal from the silent to the active state. The parameters T, N_1, N_2, and L were modified for one-minute recordings made by eight different speakers, where N_1 and N_2 are the number of times a signal

must exceed the threshold in the silent and active states to declare a sector active. The following parameters were judged to provide the largest percentage of silent intervals without clipping words or inserting silent intervals within words:

T	−40 dbm
N_1	8 out of 32
N_2	4 out of 32
L	2

This set of parameters is being used in the present system and appears to be adequate for the rate conversion and editing procedures.

The rate conversion techniques investigated retain or eliminate entire 1/16th-second sectors dependent upon a parameter which is calculated as the blocks are received. By limiting the algorithms in this manner, a decision whether or not a sector will be played back can be made before the sector is read from the disk and the playback rate can be increased without significantly increasing the number of disk accesses. Two classes of rate conversion algorithms are those which operate on silent intervals and those which operate on active speech regions. Expanding silent intervals appears to be an adequate mechanism for slowing down a message. This allows the time it takes to play back a message to be increased as much as desired without affecting the intelligibility of the words in the message. This mechanism provides a useful way to accurately position edit pointers.

Eliminating silent intervals speeds up the playback rate from 25 to 50 percent for most users, so that a one-minute message can be played back in 30 to 45 seconds. The actual rate achieved depends upon the individual speaker and whether the message is being read or created spontaneously. In the present implementation, a 1/8-second pause is inserted whenever a pause greater than or equal to 1 second occurs in the message. This appears to be sufficient to prevent different ideas or sentences in the recorded document from running together. This type of rate increase does not affect the intelligibility of individual words within the active speech regions.

The second technique investigated to increase the rate eliminates most of the silent intervals, as in the previous case, and also eliminates up to half the active blocks. It has been found that eliminating every other 1/16th second active interval creates an extremely choppy and virtually unintelligible playback. However, if only those intervals with less energy than the short-term average energy are eliminated and no more than one interval is eliminated in succession, a reasonably intelligible rendition of the message results. In the current implementation, the number of times the sampled signal exceeds a threshold during the ith interval, N_i, is used as an indication of the energy in the

signal. The moving average value of N_i at interval i, E_i, is calculated as:

$$E_i = aE_{i-1} + (1 - a)N_{i-1},$$

where

$$0 < a \leq 1.$$

A number of values of a were tried. As a is made smaller, the average tracks the changes in signal level more quickly. A value of 7/8 was found to work well in this application. If the number N_{i-1} is less than E_{i-1}, the interval is a candidate for removal and will be removed if the previous interval has not been. N_{i-1} is then used to modify the moving average in the calculation of E_i.

This rate conversion technique is also speaker-dependent. In most instances, it results in a rate increase between two and three times, so that a message that takes one minute to record can be played back in between 20 and 30 seconds. The playback resulting from this technique contains noticeable distortion, but is intelligible. It has the characteristic that those words which the speaker considered most important and spoke louder are virtually undistorted, whereas those words that were spoken softly are shortened. After a few seconds of listening to this type of speech, listeners appear to be able to infer the distorted words and obtain the meaning of the message. Extensive tests on comprehension of messages using this technique have not been conducted. However, from tests on other rate conversion techniques,[6] it is expected that comprehension will not be as good as it is at normal playback rates. This type of rate conversion is useful for users of a message system to scan a large number of messages and determine which they wish to listen to more carefully or for users of a dictation system to scan a long document to determine the areas they want to edit.

When this system was first constructed, silent intervals were played back as zero level samples. This created noticeable gaps between words. To reduce this effect, a noise signal must be inserted in these gaps. However, the background noise in recorded messages varies. To closely match the noise during silent intervals to the background noise, actual recordings of this noise are played back. In the present implementation of the system, two successive quiet intervals must be detected to make the transition from an active to a silent state. Therefore, many of the last segments of active intervals will contain only background noise. The active segment with the smallest energy is assumed to be background noise, and this interval is played back repeatedly during silent segments in the message.

VI. CONCLUSION

A speech storage system has been implemented in which a user controls speech functions with a text terminal. The operation of this system is separated between a general-purpose, time-shared computer and a special purpose minicomputer. The minicomputer transfers speech samples between a disk storage device and a telephone line and performs operations on the speech samples. The time-shared computer interfaces the user and services and controls the operations performed by the minicomputer. This functional division makes it possible to modify the user interface and services without being concerned with the real-time processing requirements of speech storage and makes it possible to merge speech storage with existing and proposed text services.

Associating a text entry and display device with the speech storage facility makes it possible to perform speech functions which are otherwise intractable. It is now possible to scan long speech messages in ways which are analogous to scanning text documents and to edit speech messages.

Three techniques for scanning speech messages have been found. Text descriptors can be associated with points in a speech message. These pointers can be listed and the speech message played back starting at a selected pointer. This is analogous to using the index in a text document to determine where to start reading. While playing back a speech message it is possible to jump forward or backward in the message. This is analogous to flipping through the pages in a text document to determine the area of interest. Finally, the playback rate can be increased. When the highest playback rate is selected, not every word is intelligible; however, the meaning can generally be extracted. This is analogous to skimming through a text document to determine the areas of interest in the document.

An editor has been implemented that allows a user to insert, delete, change, or reposition speech segments in a speech message. The editing changes are implemented at pointers in the speech message. These operations are made tractable by associating text descriptors with the pointers and locating the pointers on a time-line. The time-line is a visual display which shows the relative positions of the pointers in the speech message. To make the placement of pointers less critical of the user's and the time-shared computer's response time, these pointers are only placed at silent intervals in the speech message. And after a pointer is placed, it can be moved forward or backward in the speech message to another silent interval.

VII. ACKNOWLEDGMENTS

I wish to thank H. G. Alles for his suggestions on modifying the DMA used in this system, J. C. Schwartzwelder for implementing the re-

quired hardware configuration, and J. O. Limb and R. B. Allen for their suggestions on human interfaces and services.

REFERENCES

1. J. G. Williams, W. T. Hartwell, and G. D. Bergland, "Basic Capabilities and Future Possibilities of Centralized Network Services," ICC '79 Conference Record, pp. 3.1.1–3.1.5.
2. R. J. Nacon and D. P. Worall, "New Custom Calling Services," ICC '79 Conference Record, pp. 3.2.1–3.2.5.
3. R. G. Cornell and L. D. Whitehead, "A Centralized Approach to New Network Services," ICC '79 Conference Record, pp. 3.3.1–3.3.7.
4. R. W. Lucky, "A Flexible Experimental Digital Switching Office," Proc. 1978 International Zurich Seminar on Digital Communications, pp. A.4.1–4.4.
5. Miedema and Schachtman, "TASI Quality Effects of Speech Detectors and Interpolation," B.S.T.J. *41*, No. 4 (July 1962), p. 1453.
6. T. J. Sticht, "Failure to Increase Learning Using the Time Saved by Time Compression of Speech," J. Ed. Psych., 1971, *62* (No. 1) pp. 55–59.